

CPP程式設計題

命題者 ZZY

題目名稱(中文/英文) Design Polynomial Class

主要測試觀念：

Basics

- C++ BASICS 1
- FLOW OF CONTROL
- FUNCTION BASICS
- PARAMETERS AND OVERLOADING
- ARRAYS
- STRUCTURES AND CLASSES
- CONSTRUCTORS AND OTHER TOOLS
- OPERATOR OVERLOADING, FRIENDS, AND REFERENCES
- STRINGS
- POINTERS AND DYNAMIC ARRAYS

Functions

- SEPARATE COMPILATION AND NAMESPACES
- STREAMS AND FILE I/O
- RECURSION
- INHERITANCE
- POLYMORPHISM AND VIRTUAL FUNCTIONS
- TEMPLATES
- LINKED DATA STRUCTURES
- EXCEPTION HANDLING
- STANDARD TEMPLATE LIBRARY
- PATTERNS AND UML

題目說明： Using dynamic arrays, implement a polynomial class with polynomial addition, subtraction, and multiplication.

Remarks: A variable in a polynomial does nothing but act as a placeholder for the coefficients. Hence, the only interesting thing about polynomials is the array of coefficients and the corresponding exponent. Think about the polynomial

$$x^2 + x + 1$$

Where is the term in x^2 ? One simple way to implement the polynomial class is to use an array of doubles to store the coefficients. The index of the array is the exponent of the corresponding term. If a term is missing, then it simply has a zero coefficient. There are techniques for representing polynomials of high degree with many missing terms. These use so-called sparse matrix techniques. Unless you already know these techniques, or learn very quickly, do not use these techniques.

Please

- (1) provide a default constructor, a copy constructor, and a parameterized constructor that enables an arbitrary polynomial to be constructed.
- (2) Supply an overloaded operator = and a destructor.
- (3) Provide these operations:
polynomial + polynomial, constant + polynomial, polynomial + constant,
polynomial - polynomial, constant - polynomial, polynomial - constant.
polynomial * polynomial, constant * polynomial, polynomial * constant,
- (4) Supply functions to assign and extract coefficients, indexed by exponent.
- (5) Supply a function to evaluate the polynomial at a value of type double.

You should decide whether to implement these functions as members, friends, or standalone functions.

NOTES:

The default constructor creates an empty polynomial. A zero polynomial has degree 0, since it has only the zero degree coefficient.

In the coefficient array, the index is the value of the exponent of term having this coefficient. For example, the index 0 entry is the constant coefficient, the index 1 entry is coefficient of the linear term, the index 2 entry is the coefficient of the quadratic term (term in x^2), etc.

The size of the coefficient array include a degree 0 entry, so the size is the degree of the polynomial + 1.

Please use the following code snippets to test your program.

輸入說明： Replace the main from input-main.cpp and test your class is correct.

輸出說明： See the Sample Output

IO範例：

Sample Input

Sample Output

第一組測資與輸出
input-main1.cpp

```
Polynomial q
term with degree 0 has coefficient 3
term with degree 1 has coefficient 2
term with degree 2 has coefficient 1
Polynomial c
term with degree 0 has coefficient 1
term with degree 1 has coefficient 2
term with degree 2 has coefficient 0
term with degree 3 has coefficient 3
value of q(2) is 11
value of p(2) is 11
value of r(2) is 29
value of c(2) is 29
value of (q + c)(2) is 40
value of (q - c)(2) is -18
size of q*c is 6
Polynomial r (= q*c)
term with degree 0 has coefficient 3
term with degree 1 has coefficient 8
term with degree 2 has coefficient 5
term with degree 3 has coefficient 11
term with degree 4 has coefficient 6
term with degree 5 has coefficient 3
value of (q * c)(2) is 319
value of q(2) is 11
value of p(2) is 11
value of r(2) is 11
value of c(2) is 29
value of (q + empty)(2) is 11
value of (c - empty)(2) is 29
size of q * empty is 0
Polynomial r (= q * empty)
Polynomial q + q
term with degree 0 has coefficient 6
term with degree 1 has coefficient 4
term with degree 2 has coefficient 2
Polynomial c - c
term with degree 0 has coefficient 0
term with degree 1 has coefficient 0
term with degree 2 has coefficient 0
term with degree 3 has coefficient 0
Polynomial empty * empty
```

第二組
input-main2.cpp

Polynomial $q * q$
term with degree 0 has coefficient 36
term with degree 1 has coefficient 48
term with degree 2 has coefficient 40
term with degree 3 has coefficient 16
term with degree 4 has coefficient 4
value of $(q * c)(2)$ is 0

附屬資料：

☑解答程式：DesignPolynomialClass.cpp(檔名)

☑測試資料 input-main1.cpp, output1.txt, input-main2.cpp, output2.txt

□ 易，僅需用到基礎程式設計語法與結構

■ 中，需用到多項程式設計語法與結構

□ 難，需用到多項程式結構或較為複雜之資料型態或結構

解題時間：30分鐘。

其他註記：

***For your reference, a sample Polynomial class is as follow.

```
class Polynomial
{
public:
    Polynomial(); // creates an empty polynomial
    Polynomial(const Polynomial& poly);
    Polynomial(double coefficient[], int size);
    ~Polynomial();
    //Use indexed polynomial as r-value to inspect coefficient
    //and as l-value to assign coefficient
    double& operator[](int degree);
    //This is required if we are to have const correctness
    const double& operator[](int degree) const;

    const Polynomial& operator=(const Polynomial & rhs);
    int mySize();

    //friend functions:
    friend double evaluate(const Polynomial& poly, double arg);
    friend Polynomial operator+(const Polynomial& lhs, const Polynomial& rhs);
    friend Polynomial operator-(const Polynomial& lhs, const Polynomial& rhs);
    friend Polynomial operator*(const Polynomial& lhs, const Polynomial& rhs);
private:
    double *coef;
    int size;
};
```